ReMI
Reliability in
Mechatronics and ICT

# Embedded software ontwikkeling

Hoe passen de principes uit software engineering hierin?

**dr. ing. Jeroen Boydens**

Innovatiecentrum West-Vlaanderen

**Research @ breakfast**
**19/03/2013**

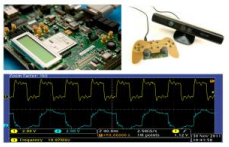
## Contents

# 1) EP research: part of ReMI Reliability in Mechatronics and ICT

**ECOREA**

ECOREA — Electronic Ciruit Design And Realization

- Analog and Digital signal processing
- Algorithm development
- FPGA & DSP Implementations

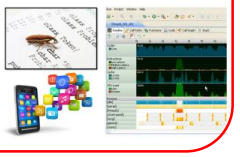


» Visit the ECOREA website

**EP**

EP — Enterprise Programming

- Concurrent software (MultiCore)
- Embedded software development
- Software testing
- Design patterns
- Software for smart phones



» Visit the EP website

**EMAS**

EMAS — Electric Machines

- Power electronics & Automation
- Power Quality (EMC)
- Efficiency & Productivity
- Reliable Networking
- Renewable Energy

**FMEC**

FMEC — Flanders Mechatronics Engineering Centre

- EMC & Signal Integrity
- overall reliability
- Thermal & Mechanical behavior
- Highly Accelerated Lifetime Testing

» Visit the FMEC website

3

# 1) EP research: part of ReMI

Founded: 2006 – 2007

**dr. ing. Jeroen Boydens**
+ founder
+ instructor
+ project leader

@ep_research

http://ep.khbo.be

**ing. Robbie Vincke**
+ MC4ES

**ing. Sille Van Landschoot**
+ MC4ES
+ TDD4ES
+ academic staff

**ing. Nico De Witte**
+ MC4ES
+ academic staff

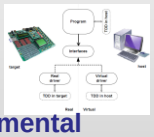
**ing. Piet Cordemans**
+ PhD student
+ TDD4ES
+ academic staff

**ing. Wim Catteeuw**
+ APS4ES

KU LEUVEN khbo

4

## EP research: threefold mission

| Education | Research | Social services |
|---|---|---|
|  |  |  |
| **Software courses**<br>➢ Prof. Bachelor EO/ICT<br>➢ Ac. Bachelor EO/ICT<br>➢ Master ICT | **Fundamental**<br>➢ Concurrent software<br>➢ Software testing<br>➢ Transactional behavior<br><br>**Applied**<br>➢ IWT/Tetra, theses<br>➢ Embedded development methodologies<br>➢ Embedded multi-core | **Seminars**<br>➢ Software testing<br>➢ State-of-the-art technologies<br><br>**LED**<br>➢ eg. Medical Reminder System<br>➢ Internal projects |

KU LEUVEN khbo

5

## Contents

1) EP research: part of ReMI
2) **TDD4ES Research overview**
3) MC4ES Research overview
4) FS4ES Research proposal
5) Workshop: Testgedreven ontwikkelen van Embedded Software – 28/05/13

KU LEUVEN khbo

6

## 2) TDD4ES research overview

### Literature study

| Icon | Rating | Title | Mindmap |
|---|---|---|---|
| 10/01/05 | 4 | Design Patterns: Elements of Reusable Object-Oriented Software | |

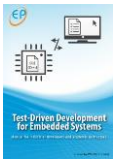
### Case studies


### Workshop

**CMS**

http://ep.khbo.be/TDD4ES

### Manual




### 3 strategies

*Test on target*

*Test on host*

*Remote testing*


KU LEUVEN khbo



## User committee TDD4ES IWT/090191

E.D.&A. ELECTRONIC CONTROLLERS
www.edna.eu

Q-STAR TEST
The Current Test Company
www.qstar.be

tesco testing your software
www.tesco.be

UNITRON GROUP
www.unitron.be

marelec IN BALANCE WITH YOUR NEEDS
www.marelec.com

Summa
www.summa.be

televic Providers of reliable interaction
www.televic.com

VAN DE WIELE
www.vandewiele.com

Newtec SHAPING THE FUTURE OF SATELLITE COMMUNICATIONS
www.newtec.eu

sirris
www.sirris.be

DSP VALLEY THE EMBEDDED TECHNOLOGY NETWORK
www.dspvalley.com

KATHOLIEKE UNIVERSITEIT LEUVEN
www.kuleuven.be

FMTC
www.fmtc.be

KdG
www.kdg.be

iwt
www.iwt.be

KU LEUVEN khbo

# Problem statement

Two trends influence the way embedded software is developed

1) *Embedded system level*:
Embedded software plays a more important role

2) *Global level*:
Embedded systems are becoming more pervasive in our lives (even in critical domains)
- Consumer electronics
- Automotive / transportation
- Medical applications

**Risk = Importance x Chance of failure**

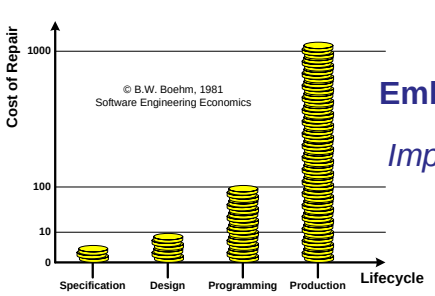
2 1

KU LEUVEN khbo

9

# Embedded quality assurance

Quality assurance in embedded software development is mostly limited to debugging and final testing, only focusing on the current issue…





**Embedded software testing gap**

*Importance* versus *state* of testing

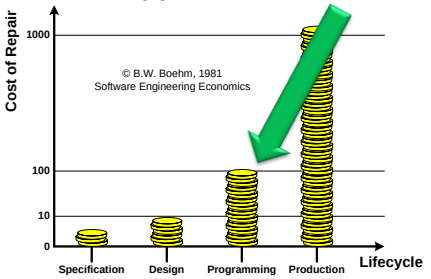
KU LEUVEN khbo

10

# Automated testing

**Timely detection of bugs by automated testing**

- Running test suite frequently during development
- Incrementally expanding the test suite
- Support from unit testing framework



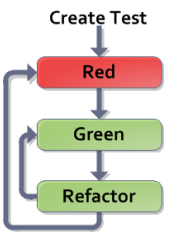

1. Find bugs early
2. Measurable quality
3. Increasing confidence
4. Detecting regression
5. Encapsulating third party code

KU LEUVEN khbo

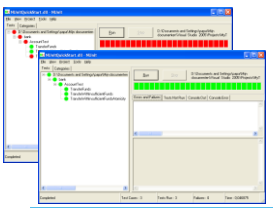
11

# Test-Driven Development

**TDD cycle**




1. Write failing test (red)
   - New behavior
   - Minimal skeleton to get through compilation
2. Write code to pass test (green)
   - Minimal implementation
3. Refactor
   - No new behavior
   - Clean code
   - Keep tests passing



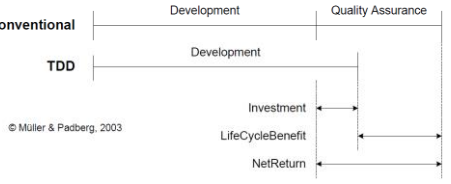
KU LEUVEN khbo

12

## Test-Driven Development

### Advantages

1. Code is tested while it's written
2. Fast feedback cycle
3. Extensive & safe refactoring
4. Focus on current functionality
5. Tests become living documentation




### Disadvantages

1. 2x code

   increase development time



## Test framework

### Why?

- Organization
- Execution
- Reuse
- Reporting

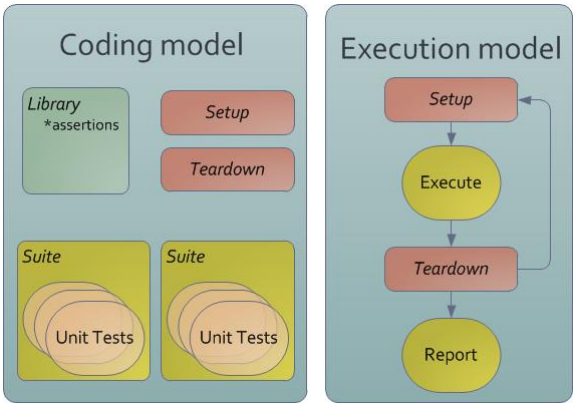
### Many exist for C/C++ (>40)

minUnit, CppUnitLite, UnitTest++, GoogleTest, CxxTest, EmbUnit, CppUTest, CppUnit, Unity…

Or: write your own (TDD-style)

# xUnit test framework

## General model






# TDD strategy

## Write test first

- Inspiration (test list)
  - Specifications
  - Use cases
  - Notes
  - …
- Procedure
  1. Define interface
  2. Select values for test

« By writing a test before implementing the item under test, attention is focussed on the item's interface and observable behavior »
Beck (2000)

## TDD tests

**Black box tests vs. White box tests**

**a.k.a. glass box tests**

- External perspective
- Specification
- Focus on end-user
- 80% coverage
- Test First

- Internal perspective
- Implementation
- Focus on developer
- 20% coverage
- Test Afterwards

17

## In conclusion

**Testing & Embedded co-design**
about changing requirements

- Broken window syndrome
  - One broken test, which remains unresolved, leads quickly to many broken tests.
- Test cancer [Fowler]
  - Removing broken tests is a bad habit.

**Maintaining tests is necessary**

18

## Embedded TDD challenges

| TDD | | Embedded |
|---|---|---|
| 1. Tests + Program + Unit test framework | ⟺ | 1. Limited resources |
| 2. Automated testing | ⟺ | 2. Hardware dependencies |
| 3. Fast programming cycle | ⟺ | 3. Slow programming cycle |



KU LEUVEN khbo

19

## Embedded constraints

↔ **Embedded TDD challenges**

1. Limit the memory footprint needed for TDD on target hardware

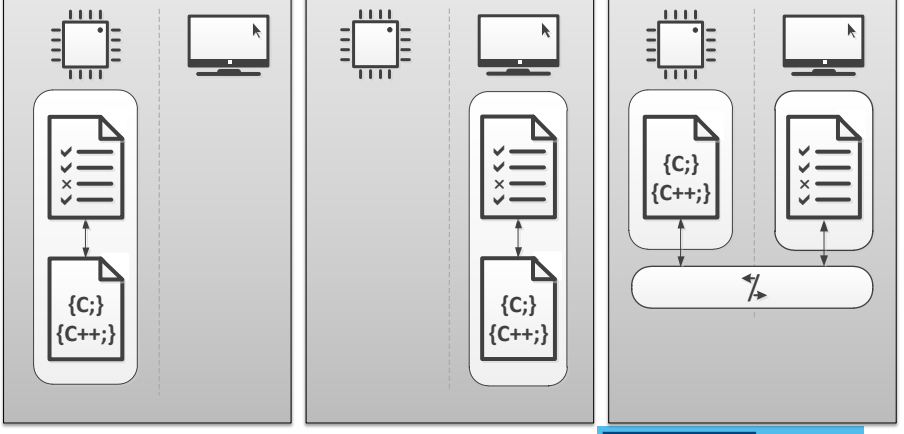
2. Tests for
   1. Hardware independent code
   2. Hardware aware code
   3. Hardware specific code

3. Maintain a fast programming cycle

KU LEUVEN khbo

20

## TDD strategies for embedded




KU LEUVEN khbo

21

## TDD strategies for embedded

**Test on target**

1. No cross-compiling

**Test on host**

1. Unlimited number of tests
2. Full option framework
3. Program code with hardware virtualization

**Remote testing**

1. Unlimited number of tests
2. Full option framework with remote testing functionality
3. Program code with remote testing functionality

KU LEUVEN khbo

22

## Evaluation: Test on Target

Advantages

- No cross-compiling

Disadvantages

- Isolating hardware behavior typically enlarges test cases (mocks)
  - Processing power
  - Memory footprint
- Slow development cycle
  - Frequent target programming

KU LEUVEN khbo

EP

24

## Test on Host: Mock verification

1. Develop test on host
   - Mock hardware
   - Correspondence with driver on target
2. Migrate test to target
   - Mock is replaced with real hardware driver

Test passes: mock & implementation = OK

**OR**

Test fails: mock & implementation != OK

KU LEUVEN khbo

EP

26

## Evaluation: Test on Host

Advantages

- Fast programming cycle
- Hardware does not need to be available
- Migration supported by test suite

Disadvantages

- Verification on target needed
  - Cross-compilation issues
  - Mock driver verification



## Remote testing

**Background**

- Inter Process Communication (IPC)
- Remote procedure call (RPC)
- Common Object Request Broker Architecture (CORBA)

**Operation**

- Tests / framework on host
- Program / drivers on target
- Management mechanism on both

## Evaluation: Remote Testing

Advantages

- Scalable alternative for test on target (ROI)
- Fast prototyping
  - Stable driver base on target
  - On host development of higher level tiers

Disadvantages

- Remote testing system
  - Development
  - Overhead: memory / processing power



## TDD4ES strategy evaluation

| | Test on target | Test on host | Remote testing |
|---|---|---|---|
| Resources  | --- Test & Program on target | +++ Test & Program on host | +/- Broker on target |
| Dependency  | +++ Real drivers | --- Only virtual drivers | +++ Real drivers |
| Slow  | --- Frequent target programming | +++ Development limited to host | - Frequent target programming* |

# Literature


- K. Beck. **Test-Driven Development: By Example**. Addison-Wesley, 2003.
- K. Beck and C. Andres. **Extreme Programming Explained: Embrace Change** (2nd ed.). Addison-Wesley, 2004.
- B.W. Boehm. **Software Engineering Economics** (Prentice-Hall Advances in Computing Science & Technology Series). Prentice Hall PTR, October 1981.
- J. Boydens, P. Cordemans, and E. Steegmans. **Test-Driven Development of Embedded Software**. In European Conference on the Use of Modern Information and Communication Technologies, 2010.
- E. Dijkstra. **The humble programmer**. Commun. ACM, 15(10):859866, 1972.
- M. Feathers. **Working Effectively with Legacy Code**. Prentice Hall PTR, 2005.
- M. Fowler. **UML Distilled** (3rd ed.). Addison-Wesley, 2004.
- M. Fowler. **Refactoring: Improving the design of existing code**. Addison-Wesley, 1999.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides. **Design patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.
- J. Grenning. **Progress before hardware**. *Agile Alliance Newsletter*, 4:74_79, 2004.




# Literature continued


- J. Grenning. **Test-Driven Development for Embedded C**. Pragmatic bookshelf, 2011.
- M. Karlesky, W. Bereza, and C. Erickson. **Effective test driven development for embedded software**. In IEEE 2006 Electro/Information Technology Conference, 2006.
- R. Koss, and J. Langr. **Test-Driven Development in C**. In *C/C++ Users Journal*, 2002.
- M. Müller and F. Padberg. **About the return on investment of test-driven development**. In International Workshop on Economics-Driven Software Engineering Research EDSER-4, 2003.
- R. Osherove. **The Art of Unit Testing**. Manning, 2009.

# Contents

1) EP research: part of ReMI
2) TDD4ES Research overview
3) **MC4ES Research overview**
4) FS4ES Research proposal
5) Workshop: Testgedreven ontwikkelen van Embedded Software – 28/05/13

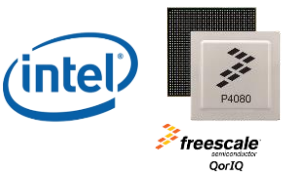
KU LEUVEN khbo



# 3) MC4ES Research overview

**Literature**


**Case Studies**


**Focus points**

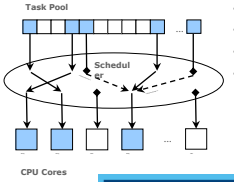
**Design patterns**

- Thread Pool Pattern
- Map/Reduce
- Loop parallelism
- Planar Pattern
- …

**Testing**

- Nonatomic operations
- Two-stage access bug pattern
- Wrong lock or no lock
- Double-checked locking
- A "blocking" critical section
- The sleep() bug
- Losing a notify
- Orphaned thread bug

**Implementation**




**Tools**

- Performance
- Safety
- Debugging
- Etc.

KU LEUVEN khbo

## User committee MC4ES IWT/110174

Alcatel·Lucent
VAN DE WIELE
TRAFICON
Altreonic
ENERGYICT
LMS
MUTOH
Atlas Copco
Cochlear
televic
FERRANTI computer systems
Atos Worldline
ONEACCESS
DANA
Newtec
BARCO
LVD
FMTC
KATHOLIEKE UNIVERSITEIT LEUVEN
Universiteit Antwerpen
imec
DSP VALLEY

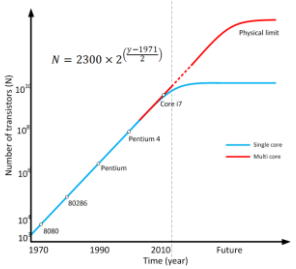
KU LEUVEN khbo

41

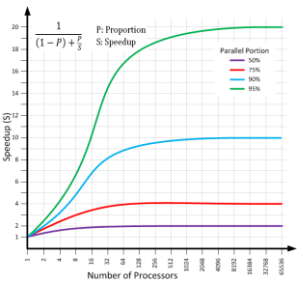
## Multicore Embedded




**Moore's Law**




**Amdahl's Law**

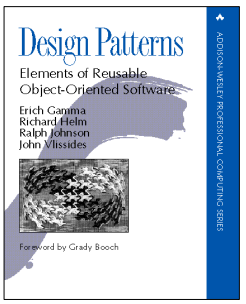



KU LEUVEN khbo

42

## Design Patterns

- A design pattern is the description of a general solution to a recurring problem [E. Gamma]



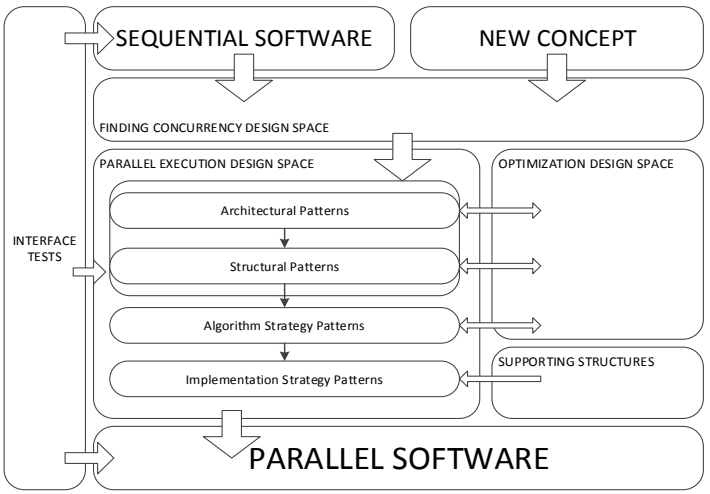

- Origin: OO-programming
- Adopted for HPC (UC Berkeley - OPL)

Embedded Parallel Software

KU LEUVEN khbo

43

## Layered Model

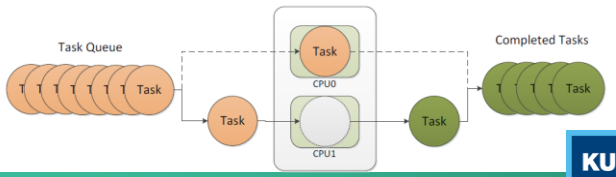



KU LEUVEN khbo

44

## Some examples …

- **Thread Pool Pattern**
  - Implementation Strategy Pattern (lower level)
- Problem? Perform parallel tasks with a static number of available units of executions
- Solution:
  - Create a static thread pool
  - Construct a shared task queue
  - Each CPU/thread can pop a task from the queue

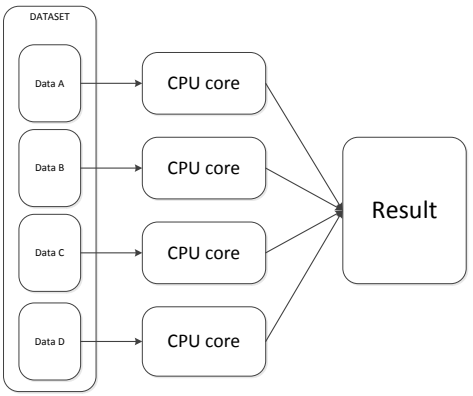



KU LEUVEN khbo

45

## Some examples …

- **Map Reduce**
  - Structural Pattern
- Problem? same operation on a large dataset
- Solution:
  - Construct a set of data-independent operations
  - Map data-chuncks to these operations
  - Reduce the sub-results to a general result

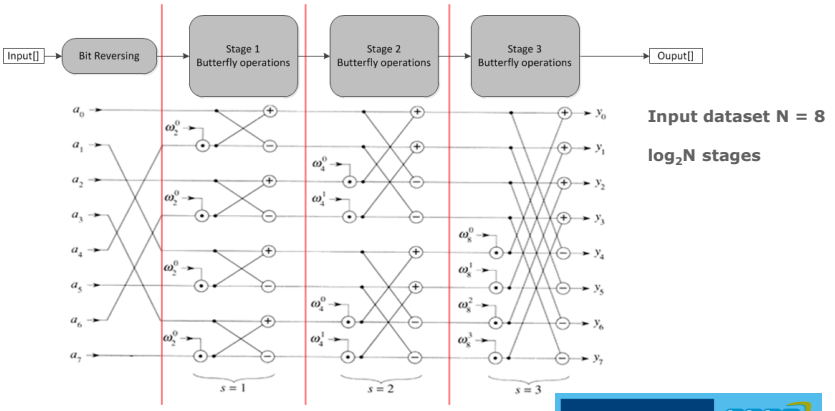



KU LEUVEN khbo

46

# Academic case study: FFT

- Radix-2 Cooley-Tukey Algorithm



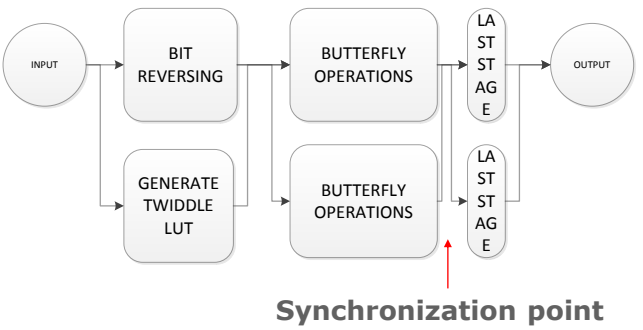

**Input dataset N = 8**

**$\log_2 N$ stages**



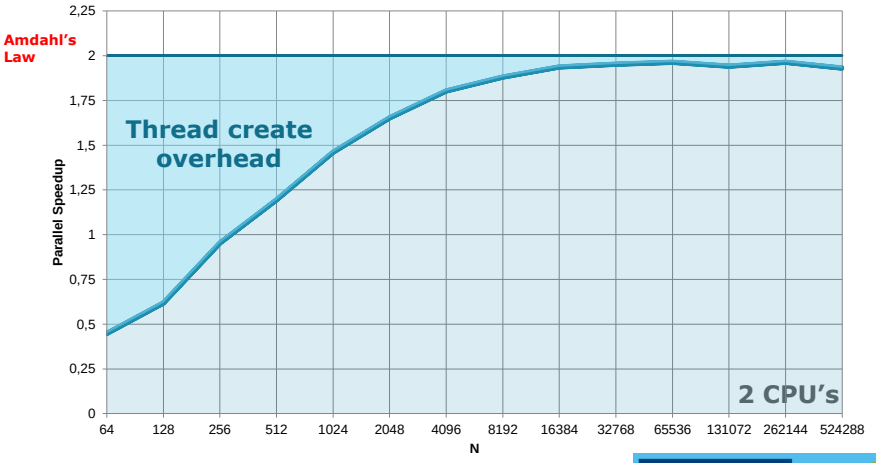
# Parallelizing: FFT

- Synchronization before last stage to avoid data races




- Parallel speedup 195%

# Measurements

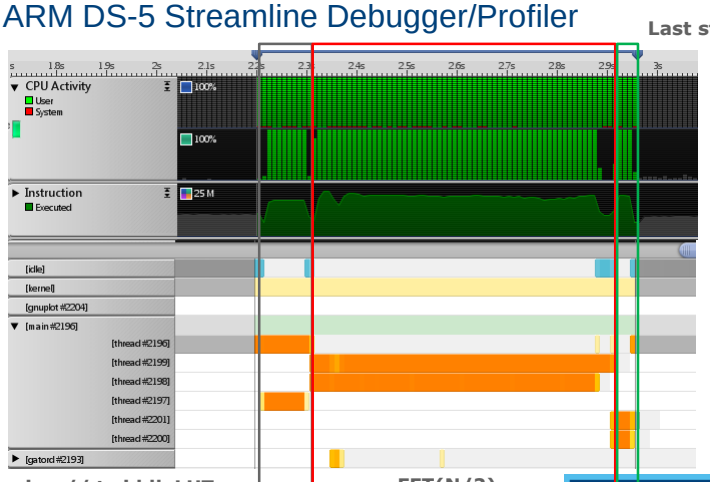



KU LEUVEN khbo

52

# Profiling Parallel code

- ARM DS-5 Streamline Debugger/Profiler




KU LEUVEN khbo

53

## Contents

1) EP research: part of ReMI
2) TDD4ES Research overview
3) MC4ES Research overview
4) **FS4ES Research proposal**
5) Workshop: Testgedreven ontwikkelen van Embedded Software – 28/05/13



## 4) FS4ES Research proposal

- **FS4ES:** Functional Safety for Embedded Software
- **Project team**:
  KHBO(KU Leuven) + KdG(UA) + FMTC
- If accepted ☺
  - 2 years, starting in October 2013
  - budget for 71 mm
- Tetra project
  - funded by IWT
  - Tetra = **Te**chnology **Tra**nsfer to improve innovation in SME in Flanders

## FS4ES Objectives

1) Design **patterns** for Functional Safety in Embedded software
2) Applying **testing techniques**, converting functional requirements into tests, simulating hardware behavior
3) Collecting **implementation guidelines**
4) Investigating **tool support**, focus on tools affordable for SME's
5) Providing **examples** in academic and industrial context

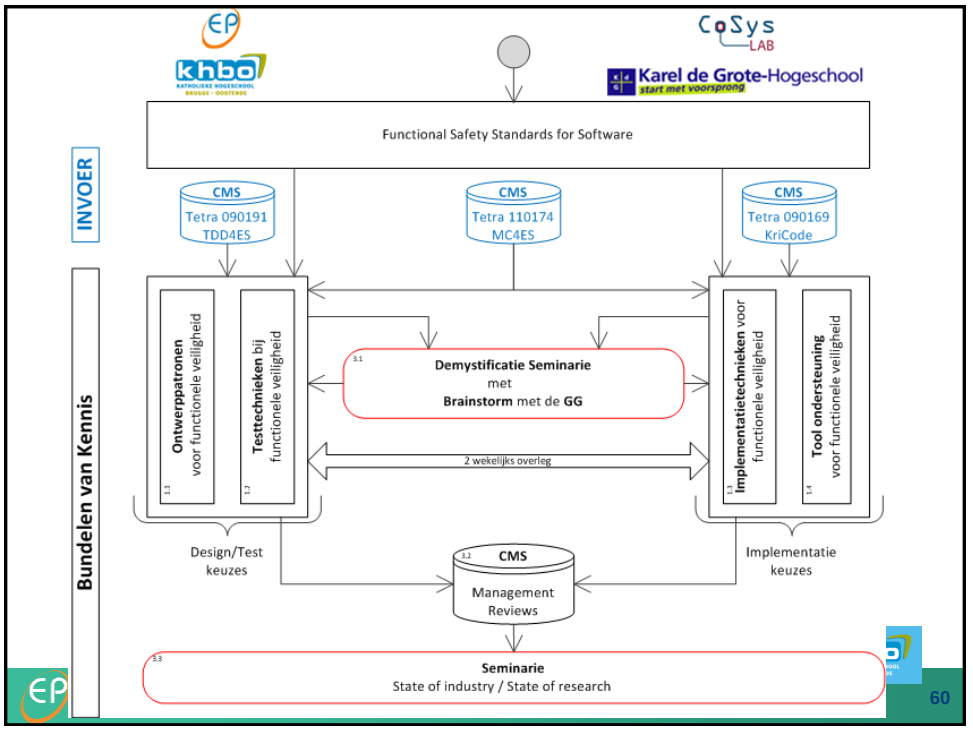
KU LEUVEN khbo

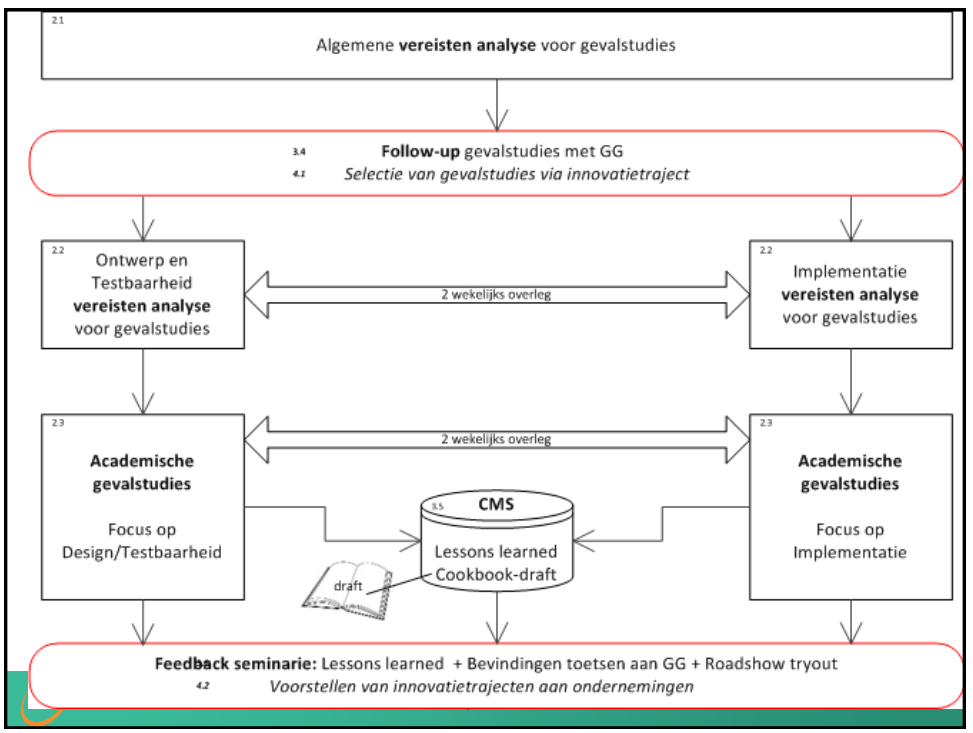


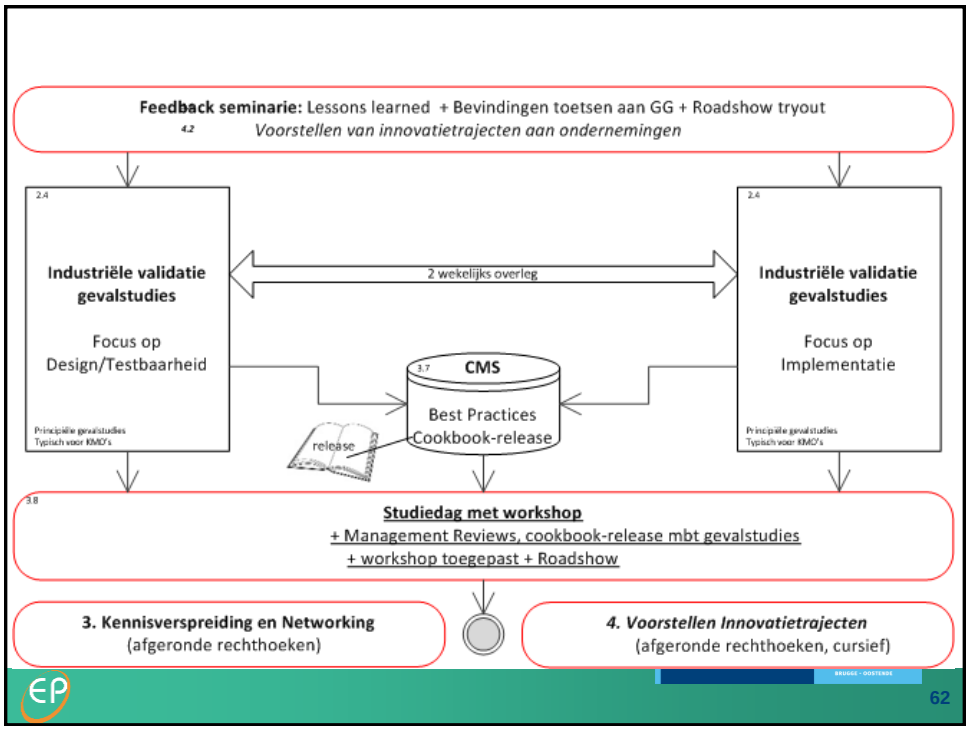
## Project User Group: determines the course of the project

- **Scientific**
  - prof. dr. ir. Eric Steegmans (KU Leuven)
  - prof. dr. ir. Yolande Berbers(KU Leuven)
  - prof. dr. Serge Demeyer (Universiteit Antwerpen)
  - Altreonic (Eric Verhulst, CTO)
  - ir Isabelle Vervenne (KHBO)
- **Valorisation**
  - DSP-Valley, FMTC, IMEC, Sirris
- **Companies**
  - Atlas Copco Airpower NV
  - Bombardier NV
  - Cochlear Benelux NV
  - Dana - Spicer Off-Highway Product Group - Dana Holding Corporation
  - E.D.&A. NV
  - EiA – Grammer
  - Emrol bvba
  - Flanders' DRIVE
  - Ferranti Computer Systems NV
  - LMS International NV
  - Melexis Technologies NV
  - QSpin Vlaanderen bvba
  - Silliberty BVBA
  - Tass Belgium NV
  - Televic RailNV
  - Teletask BVBA
  - Verhaert New Products & Services NV
- **Still open for extension!**

KU LEUVEN khbo


Functional Safety Standards for Software
INVOER
CMS
Tetra 090191
TDD4ES
CMS
Tetra 110174
MC4ES
CMS
Tetra 090169
KriCode
Bundelen van Kennis
Ontwerppatronen
voor functionele veiligheid
Testtechnieken bij
functionele veiligheid
Demystificatie Seminarie
met
Brainstorm met de GG
2 wekelijks overleg
Implementatietechnieken voor
functionele veiligheid
Tool ondersteuning
voor functionele veiligheid
Design/Test
keuzes
CMS
Management
Reviews
Implementatie
keuzes
Seminarie
State of industry / State of research
60


Algemene vereisten analyse voor gevalstudies
Follow-up gevalstudies met GG
Selectie van gevalstudies via innovatietraject
Ontwerp en
Testbaarheid
vereisten analyse
voor gevalstudies
2 wekelijks overleg
Implementatie
vereisten analyse
voor gevalstudies
Academische
gevalstudies
Focus op
Design/Testbaarheid
2 wekelijks overleg
CMS
Lessons learned
Cookbook-draft
draft
Academische
gevalstudies
Focus op
Implementatie
Feedback seminarie: Lessons learned + Bevindingen toetsen aan GG + Roadshow tryout
Voorstellen van innovatietrajecten aan ondernemingen




## Contact

- KHBO (KHBO, EP)
  - dr. ing. Jeroen Boydens
  - jeroen.boydens@khbo.be
- KdG (KdG, CoSys-Lab)
  - dr. ir. Marijn Temmerman
  - marijn.temmerman@kdg.be

63

# Contents

1) EP research: part of ReMI
2) TDD4ES Research overview
3) MC4ES Research overview
4) FS4ES Research proposal
5) **Workshop: Testgedreven ontwikkelen van Embedded Software – 28/05/13**

KU LEUVEN khbo



# 5) Workshop: Testgedreven ontwikkelen van Embedded Software – 28/03/13

## INSCHRIJVING

De deelnameprijs voor dit seminarie is € 150 per persoon. Vanaf de tweede deelnemer van hetzelfde bedrijf is de deelnameprijs € 125 per persoon. Er is een korting voorzien voor deelnemers uit de academische sector, vraag verdere inlichtingen bij Piet.Cordemans@khbo.be.

Per deelnemer wordt een PC met bijhorend embedded ontwikkelingsbord beschikbaar gesteld.

### MODALITEITEN

Inschrijven kan online op:

http://ep.khbo.be/events/...

Na inschrijving ontvangt u een factuur met overschrijvingsformulier. Het gestorte inschrijvingsgeld wordt nadien niet terugbetaald, maar indien u verhinderd bent, is het steeds mogelijk u door iemand anders te laten vervangen.

In geval van afwezigheid, wordt de syllabus per post opgestuurd.

Om een goede begeleiding tijdens de hands-on oefeningen mogelijk te maken, beperken we het aantal deelnemers. Afhankelijk van het aantal inschrijvingen kan dit seminarie meerdere keren ingericht worden.

## PRAKTISCH

LOCATIE

De cursus wordt georganiseerd door de EP onderzoekstak van de ReMI onderzoeksgroep van het department Industriële Wetenschappen en Technologie van KU Leuven campus KHBO.

Deze workshop gaat door in campus Oostende.

Adres:

**Zeedijk 101—8400 Oostende,**

toegang via Troonstraat.

### PROGRAMMA

| | |
|---|---|
| 09u00 - 09u30 | onthaal |
| 09u30 - 12u00 | eerste deel |
| 12u00 - 13u00 | lunch met belegde broodjes |
| 13u00 - 15u00 | tweede deel |
| 15u00 - 15u30 | koffiepauze |
| 15u30 - 17u00 | derde deel |

## WORKSHOP

**Testgedreven ontwikkelen van Embedded Software**

DINSDAG 28 MEI 2013

Ontvangst 09u00

ep.khbo.be

ReMI Reliability in Mechatronics and ICT

KU LEUVEN khbo

# TDD4ES

**Embedded software testen cruciaal**

Het onderdeel software vormt een essentieel deel van het ontwerp van een embedded systeem. Toch zijn er weinig technieken die zich specifiek richten op het verhogen van de kwaliteit van de broncode. Bovendien wordt embedded software vaak pas tijdens de eindfase van het ontwikkelingsproces uitvoerig getest.


**Testgedreven ontwikkelen**

Test Driven Development (TDD) is een recente techniek die beide problemen aanpakt. Deze methodologie beschouwt testen als een integraal deel van het software-ontwikkelingsproces, en biedt concrete handvaten voor het ontwikkelen van foutenvrije, goed ontworpen en goed te onderhouden software.

**Embedded obstakels**

In tegenstelling tot desktop applicaties, vereist embedded software een zeer specifieke aanpak, omdat een embedded applicatie sterk afhankelijk is van de onderliggende hardware. Testen met deze hardware zijn niet altijd mogelijk of vereisen manuele handelingen.

**4 strategieën**

In deze workshop brengen wij vier strategieën aan, die de migratie van de testgedreven principes naar een embedded omgeving mogelijk maken. Met een hands-on lab geven wij niet alleen een demonstratie van testgedreven ontwikkeling van embedded software, maar kunnen de deelnemers onmiddellijk praktische ervaring opdoen.

# PROGRAMMA

De workshop begint met een theoretische uiteenzetting over testgedreven ontwikkeling van embedded software. Onder andere de volgende vragen worden hierbij beantwoord:

- Wat zijn de voordelen van TDD?
- Wat is het red/green/refactor mantra?
- Welke tools bestaan er momenteel?
- Wat zijn de uitdagingen voor embedded?

Hierna volgen hands-on oefeningen, waarin je volgens de TDD-strategie een eenvoudige hardwareafhankelijke toepassing uitwerkt. Daarna gaan we dieper in op het toepassen van de TDD methodologie in een embedded omgeving. Hierbij passen we vier strategieën toe op een mbed ontwikkelingsbord. Op dit bord zullen we verschillende hardwarecomponenten testgedreven ontwikkelen.


Bij deelname aan dit software ontvang je de bijhorende handleiding "Test-Driven Development for Embedded Software".

## DOELGROEP

Dit seminarie is interessant voor iedere embedded softwareontwikkelaar. De hands-on oefeningen worden voorzien in C en C++, maar zijn zodanig opgebouwd dat slechts een minimale notie van OO voldoende is. Specifieke ervaring met een type microcontroller of ontwikkelings-omgeving is niet vereist.

# DOCENTEN

**dr. ing. Jeroen Boydens** is docent aan de KHBO departement Industriële Wetenschappen en Technologie. Hij doceert er vakken in het domein van software engineering en object gericht ontwikkelen. Hij is coördinator van de EP-onderzoekstak van de ReMI onderzoeksgroep van KHBO waarbinnen verschillende onderzoeksprojecten (TDD4ES, MC4ES, ...) in nauwe samenwerking met KU Leuven en de industrie lopen.

**ing. Piet Cordemans** is assistent aan de KHBO. Hij doceert er vakken rond embedded en computersystemen. Momenteel doctoreert hij rond het testen van concurrent software. In het kader van het TETRA-project TDD4ES, gesubsidieerd door IWT-090191, onderzocht hij de principes van testgedreven ontwikkeling van embedded software.

**ing. Sille Van Landschoot** is lector aan de KHBO en projectmedewerker aan de EP onderzoeksgroep. Momenteel onderzoekt hij in het kader van het TETRA–project MC4ES (IWT 110174), ontwerppatronen en testtechnieken voor multi-core embedded software. Hij was eveneens werkzaam op het TETRA-project testgedreven ontwikkeling van embedded software (TDD4ES).


66